Das TSP evolutionär bearbeitet

Definition: Traveling Salesman Problem (TSP)

Gegeben: Eine Zahl n und eine $n \times n$-Entferungsmatrix ($\in \mathbb{R}^{n \times n}$).
$\mathbb{D} = (d_{i,j})_{i,j=1,2,..,n}$, $d_{i,j} \in \mathbb{R}^+$
Gesucht: Eine Permutation $\pi = (\pi_1, \pi_2, .., \pi_n)$ von $\{1, 2, .., n\}$ mit
$(\sum_{i=1}^{n-1} d_{\pi_i,\pi_{i+1}}) + d_{\pi_n,\pi_1} =: d(\pi)$ ($\widehat{=}$ Fitness von π) ist minimal bzgl. aller Permutationen ("Optimierungsproblem").

Eigenschaften

Das Problem ist NP-vollständig. Dies gilt auch noch, wenn $\mathbb{D}$ eine euklidische Entfernungsmatrix fur n Punkte in der Ebene ist.

Der Algorithmus

Parameter und Operatoren des Algorithmus

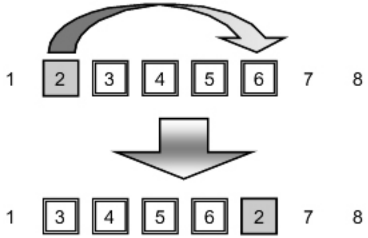
- Idividuen sind Permutationen $(\pi_1, \pi_2, .., \pi_n)$, z.B. (1,5,4,2,3).
- $\mu = 30, \lambda = 150$, Komma-Strategie
- Elternselektion $=$ Fitnessproportional
- Umweltselektion $=$ Eliteselektion (besten 30)
- Noch zu wählen
  - Mutation
  - Rekombination

Selektion

Die fitnessproportionale Selektion ermittelt für jedes Individuum der Population eine Auswahlwahrscheinlichkeit proportional zur Fitness der gesamten Population.

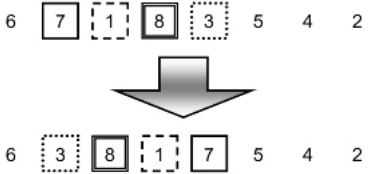
$$p(A^{(i)}) = \frac{f(A^{(i)})}{\sum_{k=1}^{r} f(A^{(k)})}$$

Mutation 1: Verschiebend



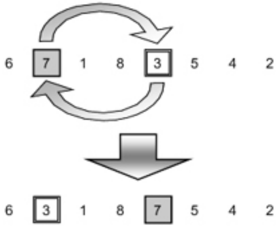
Bei Anwendung dieses Mutationsoperators wird ein zufällig bestimmtes Element aus der Permutation entfernt und an einer beliebigen Stelle wieder eingefügt. Die Zahlen, die sich zwischen der alten und neuen Position dieses Elements befinden, verschieben sich dabei um eine Stelle nach links bzw. rechts.

Mutation 2: Invertierung einer Teilsequenz



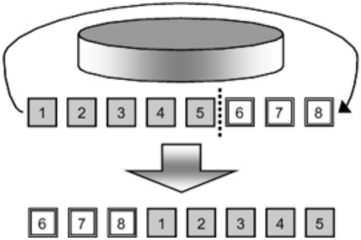
Die invertierende Mutation wählt ein zufälliges Teilstück der Permutation aus und fügt die darin enthaltenen Elemente in umgekehrter Reihenfolge wieder ein. Sei g_{perm} die Gesamtlänge der Permutation. Dann muß für die Länge l des zu invertierenden Teilstücks folgendes gelten: $2 \leq l \leq (g_{perm} - 1)$. Das obige Beispiel veranschaulicht die Wirkungsweise der invertierenden Mutation, wobei die durch Einrahmung hervorgehobenen Zahlen das zufällig ausgesuchte Teilstück der Länge 4 repräsentieren.

Mutation 3: Vertauschend



Eine sehr einfache und zudem naheliegende Möglichkeit eine Änderung an einer vorliegenden Permutation vorzunehmen, besteht darin zwei zufällig gewählte Elemente zu vertauschen. Das obige Beispiel zeigt die Anwendung dieses Operators, wobei die eingerahmten Zahlen, die durch Zufall bestimmten und zu vertauschenden Elemente kennzeichnen.

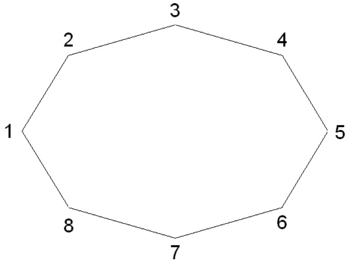
Mutation 3: Ring-Mutation



Um die Wirkungsweise der Ring-Mutation nachzuvollziehen, stelle man sich am besten einen schmalen Papierstreifen vor, der mit den Elementen der zu mutierenden Permutation beschriftet ist. Der Papierstreifen wird nun an seinen beiden Enden zu einem Ring zusammengeklebt. Dieser wird daraufhin an beliebiger Position wieder aufgeschnitten.

Vergleich der Mutations-Operatoren

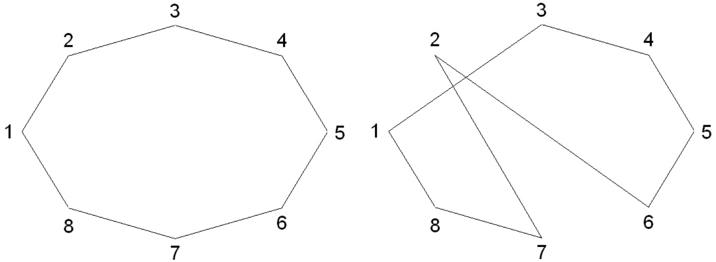
Zur Vereinfachung nehmen wir einen sehr "kleines" TSP:



Die hier dargestellte Permutation ist $(1, 2, 3, 4, 5, 6, 7, 8)$. Nun führen wir beispielhaft jeweils eine der vorgestellten Mutationen durch und betrachten welche Veränderungen (neue Kanten) dies ergibt.

Verschiebende Mutation

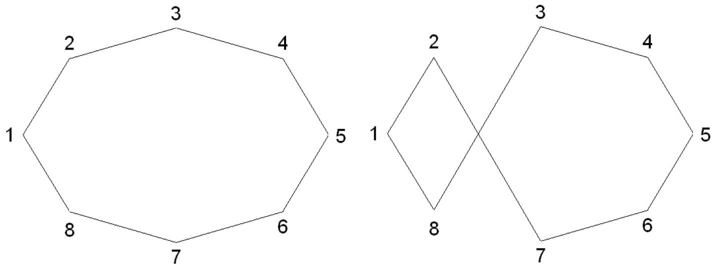
$$M_{verschiebend}((1,2,3,4,5,6,7,8),2,6) = \quad (1, {\color{red}3, 4, 5, 6, 2}, 7, 8)$$



Durch die verschiebende Mutation ergeben sich **3 neue Kanten**.

Invertierende Mutation

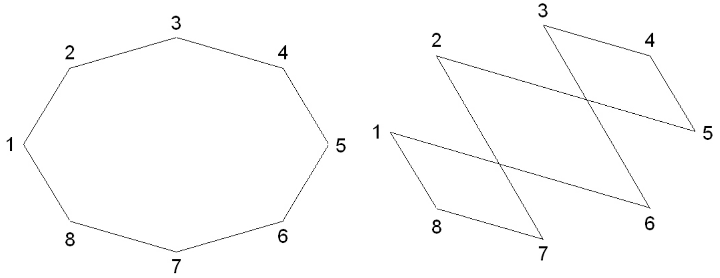
$$M_{inv}((1,2,3,4,5,6,7,8),3,7) = (1,2,7,6,5,4,3,8)$$



Durch die invertierende Mutation ergeben sich **2 neue Kanten**.

Vertauschende Mutation

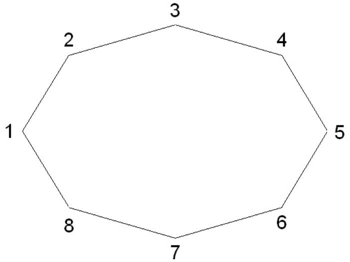
$$M_{vertauschend}((1,2,3,4,5,6,7,8),2,6) = (1,6,3,4,5,2,7,8)$$

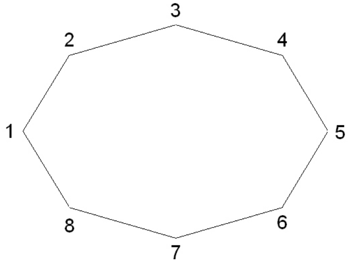


Durch die vertauschende Mutation ergeben sich **4 neue Kanten**.

Ring-Mutation

$$M_{Ring}((1,2,3,4,5,6,7,8),4) = (4,5,6,7,8,1,2,3)$$





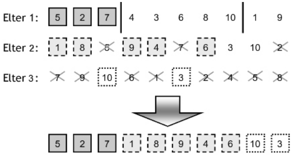
Die Ringmutation ist für das TSP gänzlich ungeeignet, da es keine relevante Änderung darstellt.

Ergebnis des Vergleichs

Mutations-Operatoren für das TSP:

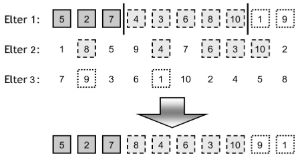
Verschiebend	3 neue Kanten
Invertierend	2 neue Kanten
Vertauschend	4 neue Kanten
Ring-Mutation	keine Änderung

Rekombination 1: n-Punkt-Crossover (streichend)



Das erste Teilstück des ersten Elters wird identisch auf das Kind vererbt. Anschließend werden alle bislang an das Kind weitergegebenen Permutationselemente im zweiten Elter durchgestrichen. Danach werden die ersten x nicht durchgestrichenen Elemente des zweiten Elters auf das Kind übertragen. Die Vervollständigung des Kindes wird dann in analoger Weise fortgesetzt. Das obige Beispiel veranschaulicht den Ablauf bei Anwendung der Rekombination mit zufälligen Teilstücken.

Rekombination 2: n-Punkt-Crossover (anordnend)



Das erste Teilstück des ersten Elters wird unverändert auf das Kind übertragen. Als nächstes werden die Elemente des zweiten Teilstücks im ersten Elter in der Reihenfolge wie sie im zweiten Elter vorkommen an das Kind weitergegeben. Für die verbliebenen Teilstücke wird analog fortgefahren.

Rekombination 3: Kantenrekombination

Man stellt sich die Permutationen als (geschlossene) Wege in einem Graphen mit der Kantenmenge $V = \{1, 2, .., n\}$ vor und versucht aus mind. zwei gegebenen Graphen mind. einen neuen Graphen zu konstruieren, der möglichst viele Kanten der "Eltern" enthält.

Der Algorithmus im Pseudocode kann dem Buch von K. Weicker ("Evolutionäre Algorithmen"), Seite 47, entnommen werden.

Rekombination 3: Kantenrekombination

Beispiel:

Es seien zwei Individuen $A = (1, 2, 3, 4, 5, 6, 7, 8)$ und $B = (1, 4, 8, 6, 5, 7, 2, 3)$ gegeben. Aus diesen Individuuen soll nun mittels Kantenrekombination ein Nachkomme erzeugt werden.

Dazu bilden wir zuerst die Adjazenten der einzelnen Knoten:

$Adj(1) = \{8, 2, 3, 4\}$ $\quad$ $Adj(2) = \{1, 3, 7\}$
$Adj(3) = \{2, 4, 1\}$ $\quad$ $Adj(4) = \{3, 5, 1, 8\}$
$Adj(5) = \{4, 6, 7\}$ $\quad$ $Adj(6) = \{5, 7, 8\}$
$Adj(7) = \{6, 8, 5, 2\}$ $\quad$ $Adj(8) = \{7, 1, 4, 6\}$

Rekombination 3: Kantenrekombination

Beispiel:

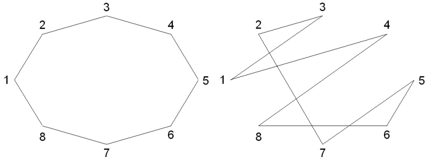
$Adj(1) = \{8, 2, 3, 4\}$ $Adj(2) = \{1, 3, 7\}$
$Adj(3) = \{2, 4, 1\}$ $Adj(4) = \{3, 5, 1, 8\}$
$Adj(5) = \{4, 6, 7\}$ $Adj(6) = \{5, 7, 8\}$
$Adj(7) = \{6, 8, 5, 2\}$ $Adj(8) = \{7, 1, 4, 6\}$

Für den Beginn des Nachkommens wählen wir zufällig den Start eines Elters. Hier beide 1.

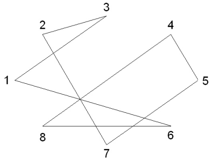
Als Nachkommen ergibt sich also schrittweise:
$Kind = (1, 3, 2, 7, 5, 4, 8, 6)$
Dabei ist die Kante $6 \longrightarrow 1$ keine Kante der beiden Eltern.

Rekombination 3: Kantenrekombination

Wir haben also aus den beiden Eltern $(1, 2, 3, 4, 5, 6, 7, 8)$ und $(1, 4, 8, 6, 5, 7, 2, 3)$



Das Kind $(1, 3, 2, 7, 5, 4, 8, 6)$ erzeugt.



Der Algorithmus

Parameter und Operatoren des Algorithmus

- Idividuen sind Permutationen $(\pi_1, \pi_2, .., \pi_n)$, z.B. (1,5,4,2,3).
- $\mu = 30, \lambda = 150$, Komma-Strategie
- Elternselektion = Fitnessproportional
- Umweltselektion = Eliteselektion (besten 30)
- Mutation = Invertierung einer Teilsequenz
- Rekombination = Kantenrekombination

Dies ist nur ein Vorschlag!